



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/016,982

Filing Date: December 14, 2001

Appellant(s): AUFFRAY ET AL.

Hunter E. Webb
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed September 23, 2008, appealing from the
Office action mailed December 27, 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,794,259	KIKINIS	8-1998
6,490,601	MARKUS ET AL.	12-2002
2002/0154162	BHATIA ET AL.	10-2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1, 3, and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kikinis (hereinafter Kikinis), U.S. Patent No. 5,794,259 issued August 1, 1998, in view of Markus et al (hereinafter Markus), U.S. Patent No. 6,490,601 filed January 15, 1999, issued December 3, 2002.

In regard to independent claim 1, Kikinis discloses filling in data on a displayed HTML form fetched from the Internet (Kikinin Abstract, column 2 lines 1-20, Figure 2).

Kikinis discloses control code in the form of a TSR program, or a plug-in module (typically downloaded) to a Web browser (a program component) (Kikinis column 3 lines 47-56).

Kikinis discloses that the plug-in is utilized for creation of bubble menus providing data to be filled in, said data pre-stored on a computer (typically contained in memory, i.e. hard drive, buffer memory, etc.). Data is then filled in the HTML form accordingly

(Kikinis Figure 2, column 3 lines 35-36, 45-55, 59-63, column 4 lines 8-25). It is noted that Kikinis teaches a Web browser (i.e. Netscape) utilizing a plug-in for implementation of its invention (Kikinis column 3 lines 49-56). Buffers for holding data were known at the time of the invention, and Netscape uses a browser cache (a type of buffer) for holding specific information. Using a typical browser cache, Kikinis's browser can check for needed data stored (or pre-stored) in its own cache. If said data is not present, it will fetch the needed data from the relevant server. It is within reason that the skilled artisan can (if he/she notices that form fields remain empty) click the "Reload" button accordingly so as to fetch data from a server (another known browser feature). Accordingly, usage of a typical browser cache buffer within Kikinis fairly teaches Applicant's claimed limitation of determining whether data is already stored in the program buffer, and filling in said data if said data is present, while fetching said data from a server if the relevant data is not in the cache.

Kikinis teaches a typical form with empty fields displayed on a client browser (Kikinis Figure 2). If needed data is not stored in the browser cache, Kikinis (via the well known use of a "Reload" button, implemented either automatically or manually), fetches data from a server. Kikinis does not specifically teach downloading said data from a server if said data is not already on the client, using said data for filling in said HTML form accordingly. However, Markus teaches filling in a form from a server, whereby a module is created on a server (privacy bank server) containing data, said module is sent to a client to be executed, resulting in data filling into said form (Markus Abstract, column 4 line 58 to column 5 line 55; compare with claim 1 "upon determination (51) that said

requested information data is not stored in the buffer memory allocated to the program component in the network client, the program component obtaining (57) the requested data by downloading the requested data from the network server and filling (59) the dedicated form fields in the hypertext document with the downloaded information data"). It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Markus to Kikinis, allowing a user of Kikinis the capability of storing fill-in data (i.e. sensitive data) off of a client computer, and on a server instead, increasing the security and "privacy awareness" of a user's information.

Kikinis discloses a user perusing a form for accuracy, to which corrections can be made prior to uploading the completed HTML form to its destination (i.e. a server) via an independently applied "Send Form" button (Kikinis column 2 lines 19-21, column 4 lines 5-9, also Figure 2 especially item 209). It is noted that, as explained above, buffers for holding data were known at the time of the invention, including browser cache, as well as input buffers for holding form input field data.

In regard to dependent claim 3, Kikinis does not specifically teach user authentication prior to display of an HTML form document. However, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Kikinis to do this, since Kikinis teaches encryption and password protected access for user profiles (Kikinis column 4 lines 32-37), providing reasonable suggestion to the skilled artisan to extend user security for entering secure sites, providing the benefit of added security to sensitive Web sites (i.e. banking sites, etc.)

In regard to dependent claim 9, Kikinis discloses filling in forms on the Internet, said forms comprising Web forms (Kikinis column 3 lines 15-30, 32-33). It is well established that Web pages on the Internet utilize the HTTP protocol (i.e. <http://www.uspto.gov>, etc.).

Claims 4-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kikinis and Markus as applied to claim 1 above, and further in view of Bhatia et al. (hereinafter Bhatia), U.S. Publication No. 2002/0154162 published October 24, 2002.

In regard to dependent claim 4, Kikinis does not specifically teach scripting (script program). However, Bhatia teaches form fill in utilizing JavaScript (Bhatia paragraphs 0057, 0088). Bhatia teaches a JAVA Web server running scripts to capture data, to process captured data, or to present processed data (paragraph [0057]). Bhatia also teaches JavaScript in paragraph [0036] to represent code assistant objects, and paragraph [0068] teaches JavaScript for verifying a host environment and at least managing the Windows registry. Paragraph [0084] teaches JavaScript associated with Internet Explorer.

In addition, please note that Bhatia paragraph [0253] teaches using JavaScript to check if a user is logged in and the page from which the user is navigating from is an ecommerce form by scanning for specific keywords in the body text (i.e. address, state, etc.). Once said form is identified, element collection is executed (triggered) accordingly.

The above, combined with the well known use of JavaScript in data collection embodiments (i.e. a JavaScript username/password box, whereby control is managed, user input is collected in field(s), said input sent to a server for validation (or validated locally), etc.), renders obvious to the skilled artisan that JavaScript can be utilized for managing form contents, triggering download/upload of stored data, accordingly. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Bhatia to Kikinis, providing Kikinis the benefit of JavaScript for increased platform independence.

In regard to dependent claims 5-8, Kikinis teaches categories of information (an identification list) (Kikinis Figure 2). Kikinis teaches a "bubble" of categories for types of data to be selected and filled in (Kikinis Figure 2 item 210). Since Bhatia teaches use of JavaScript for managing (see above), Bhatia's JavaScript can be applied to manage Kikinis's bubble selection.

Bhatia's JavaScript can also be applied to send/receive data from a server accordingly (see above). Kikinis's categories are a form of list. It would have been obvious to one of ordinary skill in the art to extend Kikinis's bubble categorization list to a record list stored on a server, so that externally saved data can be itemized and categorized accordingly (categorization can be preserved), facilitating efficient retrieval of correct data.

Kikinis does not specifically teach frames. However, Bhatia teaches HTML forms with frames (Bhatia paragraph 0076). Bhatia teaches two frames in Figure 5 (at least a

top shopping frame, and a bottom frame handling user login, etc.). Bhatia teaches a form filling service whereby each frame can contain a form (said forms can be managed using JavaScript) (Bhatia paragraph [0076]). Since both frames of Bhatia Figure 5 contain form input fields, and since both frames are related (the bottom frame controls cash back, and credit card information for the upper frame shopping site – see Bhatia paragraph [0063]), it would have been obvious to one of ordinary skill in the art at the time of the invention for Bhatia's JavaScript to be referenced within each frame, so as to facilitate coordination of data, as well as for a more pleasing visual appearance.

Bhatia teaches form fill in utilizing JavaScript (i.e. applets, etc.) (Bhatia paragraphs 0057 and 0088). It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Bhatia to Kikinis, providing Kikinis the benefit of JavaScript for increased platform independence.

(10) Response to Argument

In regard to appellant's arguments on pages 5-6, in reference to the last two limitations of the independent claims, the examiner respectfully disagrees with the appellant's position and maintains the rejection is proper. Specifically, the appellant argues that the teachings of Markus as applied in the rejection do not include uploading the modified information to the server or storing the data in the buffer memory. This assertion is used by the appellant to determine that the combination of the Kikinis and Markus references do not render the claimed invention obvious. However, the examiner maintains that the Markus reference does in fact teach uploading the modified

information to the server and storing modified information in the buffer memory. Markus discloses that bank server sends personal information to be used for automatic form filling to the client when it is requested, the user has the ability to update (modify) the information if any of it is incorrect, once the user does this the completed form is uploaded to the bank server (column 11, line 31-column 12, line 36 of Markus). Once the bank server receives the completed form it updates its raw data repository to reflect any changes the user may have made to his or her personal information (column 12, lines 17-20 of Markus).

Additionally, Markus discloses that the modified form information will be stored in buffer memory of the network client. Markus discloses that a shippable code/software module is sent to the remote computer's memory (buffer) for use by the browser in automatic form filling (column 11, lines 15-30 of Markus). The shippable code which is stored at the remote client is maintained in order for it to remain concurrent with the bank server's version of the user's personal information, thus any change in the personal information will result in an update of the shippable code which is stored at the remote client (buffer) (column 11, lines 31-62 of Markus).

These teachings provide a clear indication that Markus does in fact teach the step of uploading to the network server the modified information and storing the modified information data in the buffer memory. Thus, the examiner maintains that the invention as claimed remains obvious in view of the combination of the teachings of Kikinis and Markus as applied in the rejection above.

The appellant also argues that claims 4-8 are not obvious based on the same arguments made above in reference to claims 1, 3, and 9. Thus, the response to the arguments would be the same as the response found above.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Joshua D Campbell/

Primary Examiner, Art Unit 2178

Conferees:

/Stephen S. Hong/

Supervisory Patent Examiner, Art Unit 2178

Stephen Hong, Supervisory Patent Examiner for Group Art Unit 2178

/Doug Hutton/
Doug Hutton
Supervisory Primary Examiner
Technology Center 2100